Introduction to Swirles

and Intel GPUs in general

Juliana Kwan, 6 Mar 2025

What is Swirles?

- Swirles is our new HPC cluster and the successor to Fawcett.
- This is the official documentation for Swirles
- There are several partitions (use `sinfo` to check):
  - The Intel components are named Cosmos XI
    - `cosmos-11`: 3 nodes of Intel Data Center GPU Max 1550 (PVCs), Sapphire Rapids host with 1TB mem.
    - `universe-2`: 4 nodes of Sapphire Rapids HBMs, 1TB men
  - `ampere`: 3 nodes of Nvidia A100, host has 2TB mem
  - `lovelace`: 1 node of Nvidia L40
  - `genoa`: 6 nodes of AMD EPYC 9654, 1.5 TB mem
- Intel parts funded by ExCALIBUR as a GPU testbed
- For now, we haven't worked out a fair usage policy, expect this to change in the future

Access to Swirles

- First generate a SSH key-pair using: `ssh-keygen -t ed25519 -f <name_of_ssh_key>`

replacing <name_of_ssh_key> with something memorable. It will prompt you for a password to access the key pair, please do not leave it blank. The key pair needs to go in your ~/.ssh directory and the public key needs to be sent to help@maths.cam.ac.uk

You will only be able to login once your access has been confirmed by the Maths IT helpdesk.

- How to log in:
  - In your .ssh/config file, define a new entry:

```
Host swirles.maths.cam.ac.uk
User <username>
ProxyJump username@ssh.maths.cam.ac.uk
IdentityFile ~/.ssh/<name_of_ssh_key>
```

Then you can directly log into Swirles using `ssh username@swirles.maths.cam.ac.uk`

- Your username is your CRSid, the password is your usual Maths password for logging into your desktop or Fawcett
- How to check if you have a login (and how long your account is valid for):

  https://useradmin.maths.cam.ac.uk/selfservice/record

- If you don't have a login, you need to contact Maths IT with a support request: help@maths.cam.ac.uk Please also mention which group you belong to so that your usage can be charged correctly and so that you can be assigned the appropriate storage space.

Slurm on Swirles

How to submit jobs

- Use Slurm to queue a job with sbatch or request an interactive node with srun/salloc
- Example Slurm script (also on the GitHub repo):

```
#!/bin/bash

#SBATCH --partition <name>        #<name> can be either pvc, ampere, genoa, spr
#SBATCH --nodes=<#nodes>          #how many nodes
#SBATCH --ntasks=<#tasks>         #how many tasks (MPI ranks) in total
#SBATCH --cpus-per-task=<#cpus>   #set to > 1 to use OpenMP threads
#SBATCH --time=<hh:mm:ss>
#SBATCH --gres=gpu:<#gpus>        #necessary for GPU resources, choose from 1 - 4

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

cd ${MY_WORK_DIR}
source my_modules.sh

srun --mpi=<..> ./my_exec
```

Note that the --mpi flag needs to be set to pmi2 for Intel MPI and pmix for OpenMPI.

Slurm on Swirles

How to submit jobs

- Can also use ‘srun’ or ‘salloc’
- For example:
  - `srun -p pvc  --nodes=1 --mpi=pmi2 --cpus-per-task=28 --gres=gpu:1 - -time=00:30:00 <my_program>`
    - Run my_program on a compute node with 28 MPI ranks for 30 min but output to my current terminal using Intel MPI
  - `salloc -p pvc --nodes=1 --cpus-per-task=28 -—time=00:30:00`
    - Run an interactive job with 28 MPI ranks for 30 mins (need to ssh into the compute node yourself)
    - Alternatively, `srun … --pty bash` will redirect you automatically to a compute node
    - If using mpirun directly on PVCs, use the option --bootstrap=fork (this will expose the tiles as separate devices). Also unset ZE_AFFINITY_MASK.

The Swirles environment

- Swirles uses a module system, like on Fawcett.
- No modules are loaded by default (try doing a module list)
- To gain access to the modules compiled for Swirles type:
  - module use /cephfs/software/spack/spack-modules/maths-2024.1/linux-ubuntu22.04-x86_64_v4
  - Some essential modules:
    - module load cmake
    - module load miniforge3/4.8.3-4-Linux-x86_64/gcc/s2ikv54u
    - module load python/3.11.7/gcc/swa4n3ak
      - Then use a virtual env to install the python packages that you need.
- The OS is Ubuntu 22.04

Compilers on Swirles

- For an Intel oneAPI environment:
  - module load intel/compiler-intel-llvm/2025.0.4
  - module load intel/mpi/2021.14
  - module load intel/mkl/2025.0
  - The MPI compiler is then mpiicc/mpiicpx/mpiifx for C/C++/Fortran
- For a GNU environment:
  - module load gcc-runtime/11.4.0/gcc/3i5fbkgx
  - module load openmpi/5.0.3/gcc/xs76solb
  - The MPI compiler is then mpicc/mpicxx/mpif90 for C/C++/Fortran
- For CUDA:
  - module load cuda/12.6.3/gcc/bxp4amlg
  - The compiler is nvcc
  - Note that there is a problem using CUDA with MPI.
- NB: You can type ‘which mpirun’ to check your version of MPI

Example environment setup

You can use the script in Environments/pvc-env.sh to set up the PVCs and then test the setup using Examples/SYCL/test_sycl.cpp

The script contains:

```
#!/bin/bash

module load intel/compiler-intel-llvm/2025.0.4
module load intel/mpi/2021.14
module load intel/mkl/2025.0

export I_MPI_PMI_LIBRARY=/usr/lib/x86_64-linux-gnu/libpmi2.so  #this points Slurm to the API (PMI-2) for launching MPI jobs
export I_MPI_OFFLOAD=1                                         #this allows GPU to GPU direct communication via MPI (GPU aware MPI)
unset ZE_AFFINITY_MASK                                         #if this is set, then GPU pinning is automatically disabled
export ZE_FLAT_DEVICE_HIERARCHY=FLAT                           #exposes tiles as separate devices
export I_MPI_OFFLOAD_TOPOLIB=level_zero                        #set interface for GPU topology recognition to level zero
export I_MPI_OFFLOAD_PIN=1                                     #enable GPU pinning (somewhat degenerate with I_MPI_OFFLOAD)
export I_MPI_DEBUG=3                                           #this prints pinning information
```

This contains all the environment variables you need to run a SYCL program or use MPI enabled GPU communications or use Slurm with multiple GPUs per job

More info at Miren's guide to running GRTeclyn on Dawn: https://github.com/GRTLCollaboration/GRTeclyn/issues/67

Note that in the past I used `pti-gpu` to show the current status of the GPU but the shared filesystem that it was installed on is not mounted on Swirles. You can download and install it for yourselves here: https://github.com/intel/pti-gpu/

Storage on Swirles

Where do I put my stuff?

- Your home directory is /cephfs/home/<username> - NB: this is not the same as Fawcett.
- You should have space for outputs on /cephfs/store/<group-name> - the amount of space each group has depends on their level of contribution towards the purchase of the system.
- To check your quota use:

```
getfattr -n ceph.quota.max_bytes /cephfs/home/<your crsid>
```

Hands-on demos

- There are some simple MPI, SYCL, Pytorch scripts in the GitHub repo
  - Load your preferred modules (this must be the Intel oneAPI libraries if using SYCL)
  - Compile: mpicpx -c test_sycl.cxx
  - Link: mpicpx -o test_sycl.out -fsycl test_sycl.o
- Try out your own codes:
  - Pluto: OK
  - Athena++: OK
  - Idefix: Limited functionality, Serial host + CUDA: OK, MPI + OpenMP: OK, MPI + CUDA: Not currently possible, MPI+SYCL: OK
  - GRTeclyn/AMReX: OK, use COMP=intel-llvm (for Sapphire Rapids) or USE_SYCL=TRUE (for PVC)